

WHAT IS CLAIMED IS:

SUB A17

1. A method of inheriting data from a second dataset into a first dataset, the method comprising the steps of:

5 receiving a first user input, the first user input selecting a first data item from the second dataset for inheritance into the first dataset;

placing a first pointer in the first dataset, pointing to a first record in the second dataset that contains the first data item; and

10 when processing data in the first dataset, using the first pointer to locate the first record in the second dataset, and including the first data item from the second dataset in the processing of data in the first dataset.

2. The method of claim 1, wherein the processing of data in the first dataset is responsive to a user's selection of data items for display, and the method further comprises the following steps:

15 receiving a second user input, the second user input indicating selected data items to be displayed;

establishing a filter for identifying the selected data items to be displayed;

applying the filter to the first dataset;

20 applying the filter to the first data item, using the first pointer to locate the first data item; and

displaying data from the first dataset and from the first data item that satisfy the filter requirements.

3. The method of claim 1 further comprising the steps of:

25 receiving a third user input, the third user input indicating changes to be made to the first data item;

creating a local copy of the first data item in the first dataset; and

applying the user changes to the local copy of the first data item.

4. The method of claim 3 further comprising the step of retaining the first pointer, pointing to the first record in the second dataset.

30 5. The method of claim 3 wherein, when processing data in the first dataset, the

method includes the local copy of the first data item in the processing of data in the first dataset.

6. The method of claim 1 further comprising the steps of:
receiving a fourth user input, the fourth user input selecting a third dataset and
5 indicating that the entire third dataset is to be inherited into the first dataset;
placing a second pointer in the first dataset, pointing to the third dataset; and
when processing data in the first dataset, using the second pointer to locate the
third dataset, and including the data in the third dataset in the processing of data in the
first dataset.

10 7. The method of claim 1 further comprising the steps of:
receiving a fifth user input, the fifth user input selecting a second data item
from the second dataset for inheritance into the first dataset, wherein the second data
item has been inherited from a fourth dataset into the second dataset, the second
dataset including a fourth pointer to a second record in the fourth dataset that contains
15 the second data item;

placing a third pointer in the first dataset, pointing to the fourth pointer in the
second dataset; and

20 when processing data in the first dataset, using the third pointer to locate the
fourth pointer, using the fourth pointer to locate the second data item, and including
the second data item from the fourth dataset in the processing of data in the first
dataset.

8. The method of claim 7 further comprising the steps of:
receiving a sixth user input, the sixth user input selecting the fourth dataset
and indicating that the entire fourth dataset is to be inherited into the first dataset;
25 placing a fifth pointer in the first dataset, pointing to the fourth dataset; and
when processing data in the first dataset, using the fifth pointer to locate the
fourth dataset, and including the data in the fourth dataset in the processing of data in
the first dataset, but also detecting that the second data item has been inherited into
the first dataset both through the second dataset and directly from the fourth dataset,
30 and avoiding processing the second data item a second time.

9. The method of claim 1 further comprising a step of synchronizing the first dataset with an alter-ego dataset, including the first data item from the second dataset in the synchronization, so that after the synchronization the alter-ego dataset has a copy of the first data item from the second dataset.

10. The method of claim 9, wherein a local copy of the first data item is stored in the first dataset prior to the synchronization with the alter-ego dataset.

11. The method of claim 10 further comprising the following steps:
receiving an update to the first data item from the alter-ego dataset during the synchronization; and

entering the update from the alter-ego dataset into the local copy of the first data item.

12. The method of claim 10 further comprising the following steps:
receiving a seventh user input, the seventh user input indicating a change to be made to the first data item;

applying the user change to the local copy of the first data item;

receiving an update to the first data item from the alter-ego dataset during the synchronization;

resolving conflicts between the update to the first data item from the alter-ego dataset and the user change received in the seventh user input; and

entering the update from the alter-ego dataset into the local copy of the first data item and propagating the user change from the seventh user input to the alter-ego dataset as appropriate, based on the conflict resolution.

13. The method of claim 10 further comprising the following steps:

synchronizing the local copy of the first data item in the first dataset with the first data item in the second dataset simultaneously with the synchronization between the first dataset and the alter-ego dataset;

receiving an update to the first data item from the second dataset;

receiving an update to the first data item from the alter-ego dataset;

resolving conflicts between the updates to the first data item from the second dataset and the alter-ego dataset; and

entering the updates into the first dataset and propagating the updates to the second dataset and the alter-ego dataset as appropriate, based on the conflict resolution.

14. A method of processing data in a first dataset, the data in the first dataset including data that is native to the first dataset, data that is inherited on a record level from a second dataset, data that is inherited on a record level from the second dataset that is further inherited on a record level from a third dataset, data that is inherited on a dataset level from the third dataset, and data that is inherited from the second dataset and that is modified locally, the method comprising the steps of:

processing the data in the first dataset that are native to the first dataset;
processing the data in the first dataset that are inherited from the second dataset and for which a local copy has not already been processed; and
processing the data in the first dataset that are inherited from the third dataset and that have not already been processed during the processing of data that are inherited from the second dataset.

15. The method of claim 14, wherein the data in the first dataset further includes data that is inherited from the third dataset and that is modified locally, and the step of processing the data in the first dataset that are inherited from the third dataset excludes data for which a local copy has already been processed.

16. The method of claim 14, wherein the processing of data in the first dataset includes displaying a portion of the data from the first dataset.

17. A system for inheriting data into a first dataset from a plurality of other datasets, the system comprising:

a plurality of native data in the first dataset;
a first pointer in the first dataset, the first pointer pointing to a first data item in a second dataset to inherit the first data item from the second dataset into the first dataset on a record level;
a second pointer in the first dataset, the second pointer pointing to a third pointer in a third dataset, the third pointer pointing to a second data item in a fourth dataset to inherit the second data item from the third dataset into the first dataset on a

record level, the second data item further being inherited from the fourth dataset into the third dataset on a record level; and

a fourth pointer in the first dataset, the fourth pointer pointing to a fifth dataset to inherit the fifth dataset into the first dataset on a dataset level,

5 wherein, when the system processes data in the first dataset, the system processes data that is native to the first dataset, along with the first data item, the second data item and data from the fifth dataset.

18. The system of claim 17, wherein the second dataset and the third dataset are the same dataset.

10 19. The system of claim 17, wherein the second dataset and the fifth dataset are the same dataset.

20. The system of claim 17, wherein, if a user of the first dataset attempts to modify the first data item, the system creates a local copy of the first data item in the first dataset and modifies the local copy, instead of the first data item in the second dataset.

21. The system of claim 17, wherein, when the system processes data in the first dataset, the system detects a duplicated inheritance of a data item and avoids processing the data item multiple times.

22. The system of claim 17 further comprising a synchronizer and an alter-ego dataset, the synchronizer synchronizing the first dataset with the alter-ego dataset, including the data that is native to the first dataset, the first data item, the second data item, and data from the fifth dataset.

23. A method of inheriting data into a first dataset from one or more ancestor datasets and of synchronizing data between the first dataset and one or more alter-ego datasets, the method comprising the steps of:

25 receiving a first user input, the first user input selecting a first data item from a first ancestor dataset for inheritance into the first dataset; and

performing a first synchronization of at least a portion of the first dataset with at least a portion of a first alter-ego dataset, including sending a copy of the first data item to the first alter-ego dataset for inclusion in the first alter-ego dataset as a first alter-ego copy of the first data item.

24. The method of claim 23 further comprising the steps of:
receiving at the first alter-ego dataset a first user change to the alter-ego copy
of the first data item;
performing a second synchronization between the first dataset and the first
alter-ego dataset, including receiving at the first dataset the first user change to the
first data item;
making a local copy of the first data item in the first dataset; and
entering the first user change into the local copy of the first data item at the
first dataset.

25. The method of claim 24, wherein the step of making the local copy of the first
data item in the first dataset occurs before the first synchronization between the first dataset
and the first alter-ego dataset.

26. The method of claim 23 further comprising the steps of:
receiving at the first alter-ego dataset a second user change to the alter-ego
copy of the first data item;
beginning a third synchronization between the first dataset and the first alter-
ego dataset;
receiving at the first dataset the second user change to the first data item;
detecting a third change to the first data item in the first ancestor dataset;
making a local copy of the first data item in the first dataset;
performing a conflict resolution between the second user change to the first
data item and the third change to the first data item; and
completing the third synchronization by entering the second user change into
the local copy of the first data item, entering the third change into the local copy of
the first data item, and propagating the third change to the first alter-ego dataset, as
appropriate, based on the conflict resolution.

27. The method of claim 23 further comprising the steps of:
receiving at the first dataset a fourth user change to the first data item;
making a local copy of the first data item in the first dataset;
entering the fourth user change to the first data item into the local copy of the

first data item;

receiving at the first alter-ego dataset a fifth user change to the alter-ego copy of the first data item;

beginning a fourth synchronization between the first dataset and the first alter-ego dataset;

receiving at the first dataset the fifth user change to the first data item;

performing a conflict resolution between the fourth user change to the first data item and the fifth user change to the first data item; and

completing the fourth synchronization by entering the fifth user change into the local copy of the first data item, and propagating the fourth user change to the first alter-ego dataset, as appropriate, based on the conflict resolution.

28. The method of claim 23 further comprising the step of placing a first pointer in the first dataset, pointing to the first data item in the first ancestor dataset.

29. The method of claim 23 further comprising the step of displaying a portion of the first dataset to a user, including the first data item.